

REMARKS/ARGUMENTS

This paper is being provided in response to the July 26, 2005 Office Action for the above-referenced application. In this response, Applicant has added new Claims 77-80, and amended Claims 1, 4, 16, 19, 22, 25, 29, 30, 31, 39, 42, 54, 57, 60, 63, 67, 68, and 69 in order to clarify that which Applicant deems to be the claimed invention. Applicant respectfully submits that the newly added claims and amendments to the claims are all supported by the originally filed application.

Applicant gratefully acknowledges the indication of allowability of the subject matter of Claims 4, 6, 16-19, 25, 29, 31, 32, 34-38, 42, 44 54-57, 63, 67, 69, 70, and 72-76. Applicant has rewritten the Claims 4, 6, 16-19, 25, 29, 31, 32, 34-38, 42, 44 54-57, 63, 67, 69, 70, and 72-76 in independent form in accordance with the remarks set forth in the Office Action and respectfully submits that Claims 4, 6, 16-19, 25, 29, 31, 32, 34-38, 42, 44 54-57, 63, 67, 69, 70, and 72-76 are now in condition for allowance.

The rejection of Claims 39-75 under 35 U.S.C. 101 has been addressed by claim amendments provided herein in accordance with the guidelines provided in the Office Action. Accordingly, Applicant respectfully requests that this rejection be withdrawn.

The rejection of Claims 1-3, 5, 7-15, 20, 21, 39-41, 43, 45-53, 58 and 59 under 35 U.S.C. § 102(b) as being anticipated by Jas (U.S. Patent Publication No. 2002/0059260A1, hereinafter referred to as “Jas”) is hereby traversed and reconsideration thereof is respectfully requested. Applicant respectfully submits that Claims 1-3, 5, 7-15, 20, 21, 39-41, 43, 45-53, 58 and 59, as amended herein, are patentable over the cited reference.

Applicant's Claim 1, as amended herein, recites a method for detecting an inconsistent data structure comprising: receiving a specification describing at least one consistency constraint of a data structure; and dynamically determining during execution of a program whether said data structure violates said at least one consistency constraint, wherein said program is configured to perform said dynamically determining as part of consistency checking processing in accordance with one or more of: a signal handler that processes a detected fault, a consistency check of only a portion of said data structure at a first execution point in accordance with either a previous usage of said portion or a subsequent usage of said portion within said program, or at least one user specified execution point of said program. Claims 2, 3, 5, 7-15, 20 and 21 depend from Claim 1.

Applicant's Claim 39, as amended herein, recites a computer program product that detects an inconsistent data structure, the computer program product comprising executable code stored thereon for execution by a processor that: receives a specification describing at least one consistency constraint of a data structure; and dynamically determines during execution of a program whether said data structure violates said at least one consistency constraint, wherein said program is configured to perform said dynamically determining as part of consistency checking processing in accordance with one or more of: a signal handler that processes a detected fault, a consistency check of only a portion of said data structure at a first execution point in accordance with either a previous usage of said portion or a subsequent usage of said portion within said program, or at least one user specified execution point of said program.

Claims 40, 41, 45-53, 58 and 59 depend from Claim 39.

Jas discloses a computer-based software method for creating and modifying a database which contains a multiplicity of records consisting of attribute records, entity records, and constraint record. (See Abstract; Par. 56-62). Jas's Figure 1 includes an entity manager 102 and an entity validator 105. The entity manager 102 may present the entity validator 105 with each new and modified entity to check for consistency with the constraints before storing the new or changed entity. (Figure 1, Par. 81).

Applicant's Claim 1, as amended herein, is neither disclosed nor suggested by Jas in that Jas neither discloses nor suggests at least the features of *a method for detecting an inconsistent data structure comprising: ... dynamically determining during execution of a program whether said data structure violates said at least one consistency constraint, wherein said program is configured to perform said dynamically determining as part of consistency checking processing in accordance with one or more of: a signal handler that processes a detected fault, a consistency check of only a portion of said data structure at a first execution point in accordance with either a previous usage of said portion or a subsequent usage of said portion within said program, or at least one user specified execution point of said program*, as set forth in Claim 1. Jas discloses performing consistency checking with constraints before storing a new or changed entity. However, Jas appears silent with regard to performing consistency checking in accordance with a signal handler, a consistency check of only a portion of a data structure, or at least one user specified execution point. Accordingly, Jas neither teaches, discloses nor suggests at least the foregoing recited features of Claim 1.

For reasons similar to those set forth regarding Claim 1, Claim 39 is also neither disclosed nor suggested by Jas in that Jas neither discloses nor suggests at least the features of *a*

computer program product that detects an inconsistent data structure, the computer program product comprising executable code stored thereon for execution by a processor that: ... dynamically determines during execution of a program whether said data structure violates said at least one consistency constraint, wherein said program is configured to perform said dynamically determining as part of consistency checking processing in accordance with one or more of: a signal handler that processes a detected fault, a consistency check of only a portion of said data structure at a first execution point in accordance with either a previous usage of said portion or a subsequent usage of said portion within said program, or at least one user specified execution point of said program, as set forth in Claim 39.

In view of the foregoing, Applicant respectfully requests that the rejection be reconsidered and withdrawn.

The rejection of Claims 22-24, 26-28, 30, 33, 60-62, 64-66, 68, and 71 under 35 U.S.C. § 102(e) as being anticipated by Applin (U.S. Patent Publication No. 2004/0015876A1, hereinafter referred to as “Applin”) is hereby traversed and reconsideration thereof is respectfully requested. Applicant respectfully submits that Claims 22-24, 26-28, 30, 33, 60-62, 64-66, 68, and 71, as amended herein, are patentable over the cited reference.

Claim 22, as amended herein, recites a method of dynamically repairing an inconsistent data structure during program execution comprising: receiving at least one inconsistency violation; selecting a repair to correct said at least one inconsistency violation; and repairing said inconsistent data structure, said repairing including modifying at least a portion of said inconsistent data structure. Claims 23, 24, and 26-28 depend from Claim 22.

Claim 30, as amended herein, recites a method of handling an invalid memory reference comprising: determining whether a memory reference associated with an operation is invalid; and if said memory reference is invalid, performing an action selected in accordance with a type of said operation, wherein said type includes at least one of a read operation or a write operation.

Claim 33 depends from Claim 30.

Claim 60, as amended herein, recites a computer program product that dynamically repairs an inconsistent data structure during program execution, the computer program product comprising executable code stored thereon for execution by a processor that: receives at least one inconsistency violation; selects a repair to correct said at least one inconsistency violation; and repairs said inconsistent data structure, said repairs including modifying at least a portion of said inconsistent data structure. Claims 61, 62, and 64-66 depend from Claim 60.

Claim 68, as amended herein, recites a computer program product that handles an invalid memory reference comprising executable code stored thereon for execution by a processor that: determines whether a memory reference associated with an operation is invalid; and if said memory reference is invalid, performs an action selected in accordance with a type of said operation , wherein said type includes at least one of a read operation or a write operation. Claim 71 depends from Claim 68.

Applin discloses performing checks to verify proper pointer usage and initiate any appropriate processing to avoid a fatal error, execution of an invalid operation and/or provide notification of an error, warning, or other condition. Applin discloses checking for a pointer not

properly aligned for the type of data to be retrieved, and other pointer usage inconsistent with a particular operation, data type, etc. Applin discloses processing if a misuse or potential misuse is detected including, for example, terminating program execution and generating an error message, providing a warning message without terminating program execution, providing alternative processing and/or default values in response to identification of a pointer problem without program termination. Applin discloses determining and initiating a desired action. Applin discloses a safe or recovery state being identified and instantiated when an inappropriate use of the null pointer is attempted. (Par. 15-16; Figure 1).

Applicant's Claim 22, as amended herein, is neither disclosed nor suggested by Applin in that Applin neither discloses nor suggests at least the features of *a method of dynamically repairing an inconsistent data structure during program execution comprising: ... selecting a repair to correct said at least one inconsistency violation; and repairing said inconsistent data structure, said repairing including modifying at least a portion of said inconsistent data structure*, as set forth in Claim 22. Applin discloses, in response to pointer misuse, performing processing including supplying a default value and identifying a safe or recovery state. However, Applin neither discloses nor suggests modifying the inconsistent data structure. Accordingly, Applin neither teaches, discloses nor suggests at least the foregoing recited features of Claim 22.

For reasons similar to those set forth regarding Claim 22, Applicant's Claim 60, as amended herein, is neither disclosed nor suggested by Applin in that Applin neither discloses nor suggests at least the features of *a computer program product that dynamically repairs an inconsistent data structure during program execution, the computer program product*

comprising executable code stored thereon for execution by a processor that: ... selects a repair to correct said at least one inconsistency violation; and repairs said inconsistent data structure, said repairs including modifying at least a portion of said inconsistent data structure, as set forth in Claim 60.

Applicant's Claim 30, as amended herein is neither disclosed nor suggested by Applin in that Applin neither discloses nor suggests at least the features of *a method of handling an invalid memory reference comprising: ... if said memory reference is invalid, performing an action selected in accordance with a type of said operation, wherein said type includes at least one of a read operation or a write operation*, as set forth in Claim 30. Applin discloses, in response to pointer misuse, performing processing including supplying a default value and identifying a safe or recovery state. However, Applin neither discloses nor suggests performing any particular action for a read operation or a write operation. Accordingly, Applin neither teaches, discloses nor suggests at least the foregoing recited features of Claim 30.

For reasons similar to those set forth regarding Claim 30, Applicant's Claim 68, as amended herein, is neither disclosed nor suggested by Applin in that Applin neither discloses nor suggests at least the features of *a computer program product that handles an invalid memory reference comprising executable code stored thereon for execution by a processor that: ... if said memory reference is invalid, performs an action selected in accordance with a type of said operation, wherein said type includes at least one of a read operation or a write operation*, as set forth in Claim 68.

In view of the foregoing, Applicant respectfully requests that the rejection be reconsidered and withdrawn.

Based on the above, Applicant respectfully requests that the Examiner reconsider and withdraw all outstanding rejections and objections. Favorable consideration and allowance are earnestly solicited. Should there be any questions after reviewing this paper, the Examiner is invited to contact the undersigned at 508-898-8604.

Respectfully submitted,
MURHEAD AND SATURNELLI, LLC

 (vs 33,978)
Anne E. Saturnelli
Reg. No. 41,290

Muirhead and Saturnelli, LLC
200 Friberg Parkway, Suite 1001
Westborough, MA 01581
Tel: (508) 898-8601
Fax: (508) 898-8602

Date: October 26, 2005



IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant: Brian C. DEMSKY et al. :
Serial No.: 10/723,252 : Examiner: Bryce P. Bonzo
Filed: November 26, 2003 : Art Unit: 2114
For: SPECIFICATION BASED DETECTION : Atty. Docket: MIS-00401
AND REPAIR OF ERRORS IN DATA
STRUCTURES

Certificate of Mailing

I hereby certify that the foregoing document is being deposited with the United States Postal Service, postage prepaid, first class mail, in an envelope addressed to Mail Stop Amendment, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on October 26, 2005.



Bonny Rogers

REQUEST FOR CHANGE OF ADDRESS

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

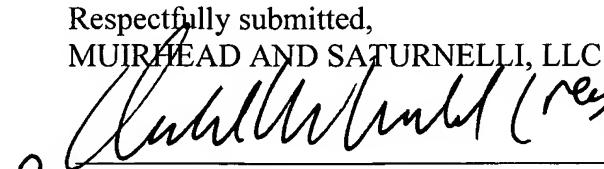
Sir:

Please direct all correspondence for this application to the following address:

Muirhead and Saturnelli, LLC
200 Friberg Parkway, Suite 1001
Westborough, Massachusetts 01581

In addition, please direct all telephone calls for this application to Anne E. Saturnelli at 508-898-8604.

Respectfully submitted,
MUIRHEAD AND SATURNELLI, LLC


Anne E. Saturnelli
Registration No. 41,290

October 26, 2005
Date

Muirhead and Saturnelli, LLC
200 Friberg Parkway
Suite 1001
Westborough, MA 01581